Гусев Максим Андреевич
Email: g.max01@mail.ru
Муромский институт (филиал) ФГБОУ ВПО «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

Парсинг, чанкинг, эмбеддинги и базы знаний: полный пайплайн превращения данных в основу для интеллектуальных систем

В контексте широкого распространения больших языковых моделей (LLM) ключевой проблемой становится не столько архитектура модели, сколько качество и структура данных, используемых для её функционирования. Значительная часть релевантной информации представлена в неструктурированном виде (документы, PDF-файлы, веб-страницы), что делает её непригодной для непосредственной обработки искусственным интеллектом. Несмотря на способность LLM генерировать связные тексты, их эффективность в прикладных задачах, таких как вопросно-ответные системы или анализ документов, напрямую зависит от точности извлечения информации из внешних источников. Невозможность обработки объемных текстов целиком из-за ограничения длины контекста обуславливает необходимость разработки специализированного конвейера предобработки данных.

Решение данной проблемы заключается в построении четкого пайплайна обработки информации. Ключевыми этапами данного пайплайна являются парсинг, чанкинг, генерация векторных эмбеддингов и интеграция с векторными базами данных. Совокупность этих технологий формирует основу для систем с расширенным поиском (Retrieval-Augmented Generation, RAG). Архитектура RAG-системы представлена на рисунке 1. Данный подход позволяет трансформировать неструктурированные текстовые массивы в структурированную, доступную для машинной обработки базу знаний.

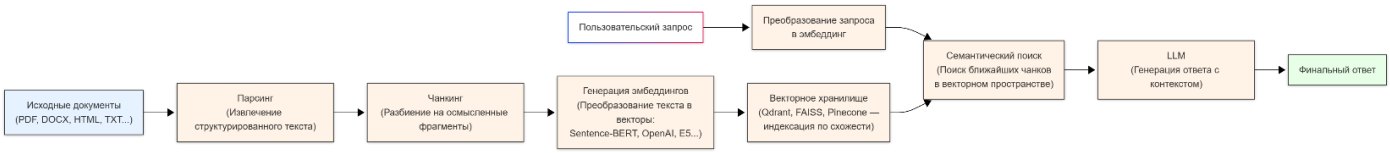

Рисунок 1 - Архитектура RAG

Исходя из рисунка 1, можно сделать вывод, что построение RAG-системы состоит из четырех последовательных этапов.

Этап 1. Парсинг исходных данных. На данном этапе производится извлечение текстового содержимого из исходных форматов (PDF, HTML, DOCX). Качество выполнения этого этапа является критически важным, так как определяет результат всех последующих операций. Задача осложняется разнородностью источников: PDF-файлы могут содержать как машинно-читаемый текст, так и сканированные изображения, требующие применения OCR-систем. Даже в текстовых PDF может происходить потеря структурных элементов документа, таких как таблицы и колонки. Для решения этих задач применяются специализированные инструменты, краткая характеристика которых представлена в таблице 1.

Таблица 1 – Инструменты для парсинга

Инструмент	Форматы	Особенность
PyPDFLoader	PDF	Быстрый, но без структуры
UnstructuredPDFLoader	PDF	Сохраняет заголовки, таблицы, колонки
BeautifulSoup	HTML	Чистит веб-страницы от шума
Apache Tika	PDF, DOCX, PPTX, HTML и др.	Универсальный парсер «всё в одном»

Этап 2. Чанкинг текста. После извлечения текста возникает задача его сегментации на семантически осмысленные фрагменты (чанки), пригодные для обработки моделью. Нативный подход, использующий разбиение по фиксированному количеству символов, часто приводит к потере смысла на границах фрагментов. Более эффективными являются методы, учитывающие структуру документа, такие как рекурсивное разделение или разделение по структурным маркерам (например, заголовкам). Важным параметром является overlap (перекрытие) – буферная зона между чанками, позволяющая сохранить контекстную связность. Сравнительный анализ методов чанкинга приведен в таблице 2.

Таблица 2 – Сравнение методов чанкинга

Метод чанкинга	Преимущества	Недостатки	Подходит для
Фиксированный размер	Простота, предсказуемость	Разворачивает предложения, теряет смысл	Технических логов, код
По абзацам/заголовкам	Сохраняет структуру документа	Может превысить лимит токенов	Статьей, отчётов, документации
Семантический чанкинг	Максимально сохраняет смысл	Требует вычислений, сложен в настройке	Юридических, медицинских текстов
Recursive (LangChain)	Гибкий, адаптивный	Может быть медленным	Универсальный

Этап 3. Генерация векторных эмбеддингов. На данном этапе текстовые чанки преобразуются в числовые представления – векторы в многомерном пространстве. Для этого используются специализированные модели эмбеддингов (например, OpenAI Embeddings, Sentence-BERT). Ключевым свойством качественных эмбеддингов является отображение семантической близости текстов в векторном пространстве: смысла схожие фрагменты имеют близкие векторные представления. Это свойство является основой для последующего семантического поиска.

Этап 4. Интеграция с векторными базами данных и оркестрация. Векторные представления чанков вместе с их метаданными индексируются в специализированных базах данных (Chroma, FAISS, Qdrant), оптимизированных для быстрого поиска ближайших соседей. Оркестрация всех компонентов пайплайна, включая семантический поиск и взаимодействие с LLM, осуществляется с использованием фреймворков, таких как LangChain. Это позволяет реализовать RAG-парадигму: пользовательский запрос преобразуется в эмбеддинг, векторная база возвращает релевантные чанки, которые вместе с запросом подаются в LLM для генерации обоснованного ответа, что позволяет минимизировать «галлюцинации» модели.

А также в рамках исследования был развернут экспериментальный ИИ-ассистент на базе рассмотренного пайплайна. В качестве тестовой базы знаний использовалась база знаний Linux Wiki. Качество работы системы оценивалось по метрикам точности извлечения фактов (Factual Accuracy) и релевантности ответов (Answer Relevance). Предварительные результаты показали, что использование RAG-подхода позволило достичь точности извлечения фактов на уровне ~92% и релевантности ответов ~88% на тестовой выборке, что подтверждает эффективность предложенной методологии для построения вопросно-ответных систем над специализированными корпусами текстов.

Таким образом, представленный пайплайн является итеративным процессом, требующим тонкой настройки каждого этапа под конкретную задачу. Однако его корректная реализация позволяет преодолеть ключевое ограничение LLM – работу с объемными неструктурированными данными – и служит основой для создания надежных и точных интеллектуальных систем.

Выражение благодарности:

Соавторы работы: к.т.н., доцент Астафьев Александр Владимирович, МИВлГУ